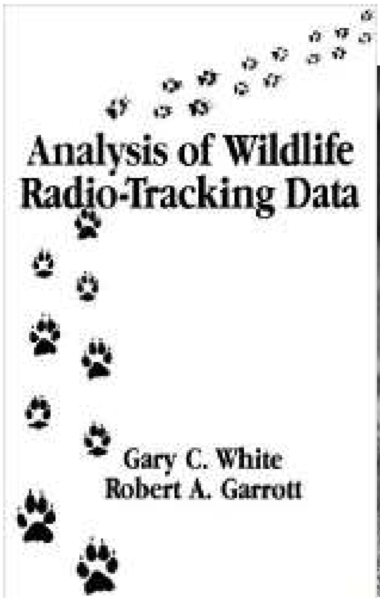
Exercise 1: The Known-Fate Model

This exercise closely follows the content of the 4th lecture and is mostly intended to show how to run program MARK to compute survival from 'known-fate' data.. We have prepared exercises for use in either MARK or RMark.



Data input

Input data consists of summarized frequencies of encounter-histories. Encounter-history is distinguished here from capture-history by having two codes for each occasion - One for capture-status, and one for recapture-status. The possibilities here are: 11=captured and dead recovery, and 10=captured and not recovered as dead.

The example we'll use for this model contains 2 groups of animals: Treatment and Control. So, each encounter-history will have 2 frequencies following the encounter-history. The following is a sample MARK input file (knownfate.inp):

```
/* known fate example (pg 344) */
10 19 21   /* 19 treatment-alive, 21 control-alive*/;
11 38 38   /* 38 treatment-dead,  38 control-dead */;
```

Explanation: The '10 19 21' means that 19 animals from group 1 (treatment) and 21 animals from group 2 (control) were captured in time 1 and were not recovered as dead (ie. they survived). The '11 38 38' means that 38 animals from group 1 and 38 animals from group 2 were captured in time 1 and recovered as dead in time 2 (ie. they did not surive).

For R users

1) Open the file "Known_Fate.r" in your preferred text editor for R (e.g., R Studio, Tinn-R, Notepad++). Note that the hash (#) symbol denotes comments in R and everything on the line after that symbol is ignored by R. The first non-comment lines in the file clear the R workspace (memory) and set the working directory(folder) to the folder where you have the exercises. You will need to modify the working directory. Execute these lines in R. (In R studio or Tinn-R, place the cursor on the first of these lines, then click the 'Run' button 3 times. In Notepad++, highlight those 3 lines, copy to clipboard, then paste into R window.)

```
rm(list=ls())                                 #   clear workspace
setwd('h:/x/workshops/uf2016/exercises')  # set working directory(folder)
library(RMark)
```

2) Assuming that you have already created a MARK input file, the first step in using RMark is to convert the input file into an RMark input "data-frame". (You can think of a data-frame as a spreadsheet in R, where you can access the spreadsheet by rows and columns, or by variable(column) name. This conversion is done with the "convert.inp" function. The only required argument is the input filename (knownfate.inp), but since we have 2 groups in our input file, we also need to specify the group names to the convert.inp function. This is done with the group.df argument. The converted data-frame is saved with the R variable name, "mdeerinp". Execute that line, then type the new variable name into the R window.

```
mdeerinp= convert.inp('knownfate.inp',group.df=data.frame(trt=c('trt','cnt')))
mdeerinp
```

3) The 2nd step is to create a processed-data variable which contains other variables needed to setup and run the MARK models. The variable, "mdeerpr" is a "list" variable, or a variable which contains other variables. The RMark function to do this is "process.data" and requires the converted input from the previous step, the type of MARK models which will be run, and the group variable name (if applicable). Execute this line, then type 'mdeerpr' to see the contents of this variable.

```
mdeerpr = process.data(mdeerinp, model='Known', groups="trt")

mdeerpr
```

4) The 3rd step is to create design-matrix data variables, needed by MARK to build models. The RMARK function is "make.design.data" and the processed-data variable created in the previous step is needed as an argument to the function. Execute this line and type 'mdeerdd' to view the contents of this variable. This is a list-type variable and it contains a data-frame variable (S) for

the estimated parameter for the model-type (Known) we specified. The columns of S are the variables we can use in building MARK models.

```
mdeerdd = make.design.data(mdeerpr)
mdeerdd
```

Modeling strategy is to develop 2 models. One represents the null hypothesis that there is no difference between the survival rates of the deer in the 2 treatments. The other model represents the alternative hypothesis that the treatment and control groups show different survival rates (control survival > treatment survival).

5) Now we're ready to build our first MARK model. With only one survival interval in the data, we don't have a lot of choices in building models. We'll start with the most simple model (survival equal among treatment and control groups). To create a model, we call the 'make.mark.model' function with the processed-data variable created in step 3 (mdeerpr), the design-data variable created in step 4 (mdeerdd), title, and list of parameters as arguments. In this case, there is only 1 parameter (S) and we specify the model using an R formula. Here, formula=~1 means the parameter is a constant value. Execute these lines and type 'mod_null' to examine what we've created. This variable is another list-variable which contains other variables to be used when we run the model.

```
Mod_null=make.mark.model(
  mdeerpr,mdeerdd,title='MuleDeerdata',
  parameters=list(
    S=list(formula=~1)
  )
)
mod_null
```

6) We can run the model by calling the 'run.mark.model' function with the model variable created in step 5 as the argument. Execute this line then type 'mod_null_out' to examine the contents. The output from MARK is stored in a text file and by typing 'mod_null_out', the text file is opened in notepad. This is the usual output you would get if you ran MARK interactively using its GUI. The output is also stored as an R list-variable. Type 'str(mod_null_out)' to display the structure of this list-variable. What is the survival rate estimate from this model? You can get it from the text output displayed in notepad, or by looking at the mod_null_out$results$real$estimate variable.

```
mod_null_out=run.mark.model(mod_null)
str(mod_null_out)
print(mod_null_out)
print(mod_null_out$results$real)
```

7) To run another model, we only need to repeat the last 2 steps: create.mark.model, and run.mark.model. With these data, the interest was in whether the treatment affected survival, so we'd like to try a model where survival is different for control and treatment groups. So, we make a new mark model, named mod_trt, where survival is different among the 2 groups (S=list(formula=~group)). Execute these lines, as well as the line to run the model (mod_trt_out=run.mark.model…). What are the survival rate estimates for the 2 groups? Are they different? Are they significantly different?

```
mod_trt = make.mark.model(mdeerpr,mdeerdd,parameters=list(
 S=list(formula=~group)
))
mod_trt_out = run.mark.model(mod_trt)
print(mod_trt_out$results$real)
```

8) We can create a table of model results by calling the 'model.table' function, with the names of the variables which contain the output of each model as an argument. Execute this line, and the next line to print the table. Assuming AIC was covered in the lecture, what does the table tell you about the effect of treatment on survival rates? Do a likelihood-ratio test between models. Does the likelihood-ratio test lead to the same conclusion as model selection?

```
#          create AIC table of model results for model comparison
tbl=model.table(model.list=c("mod_null_out","mod_trt_out"),type="Known")
print(tbl)
```

Questions:

1. Which model would you choose to best describe these data? Why?
2. Does a "z-test" comparison of survival rates between the 2 groups agree with the AIC model selection table?
    a. $Z = (S_{cnt}-S_{trt})/(var(S_{cnt})+var(S_{trt}))$
    b. Need R function, pnorm
3. Do the probability levels associated with the z-test and likelihood ratio test differ? If so, why?
    a. LR=Pr(Chi-sq, k), where Chi-sq = $L_{null}-L_{trt}$ and k=$df_{null}-df_{trt}$
    b. Need R function, pchisq

Exercise 2: The CJS Model

This exercise closely follows the content of the 5th lecture and is mostly intended to show how to run program MARK to compute survival and capture probabilities from 'capture-recapture' data.. We have prepared exercises for use in either MARK or RMark.

Data input

Background
Data for this example came from the trapping of meadow voles, *Microtus pennsylvanicus*, at Patuxent Wildlife Research Center, Laurel, MD (Nichols *et al.*, 1984). Data were collected on a 10 x 10 grid of trapping stations spaced at 7.6m intervals in old field habitat. A single modified Fitch live trap (Rose, 1973) was placed at each station. Hay and dried grass were placed in the traps and whole corn was used as bait. Sampling occurred for five consecutive days each month, from June 1981 through December 1981. During each 5day trapping session, traps were opened in the evening of the first day, checked the following morning, locked open during the day, and reset in the evening, with the sequence repeated each day until 5 days had elapsed. A racoon, *Procyon lotor* (later captured), visited the traps on the final two nights of the second trapping session, essentially leaving only 3 days of trapping for this session. At each capture, animals were examined for a tag, sexed, weighed, and examined for external reproductive characteristics. Tagged animals were ear-tagged with numbered fingerling tags, and tag numbers of marked animals were recorded at each capture.

We used 'adult' (>22g) and 'young' (<=22g) animals and collapsed the 5 days of sampling each month into a single assessment of presence or absence, leaving six monthly sampling occasions.

For R users

1) Open the file “cjs2age.r” in your preferred text editor for. Note that the hash (#) symbol denotes comments in R and everything on the line after that symbol is ignored by R. The

first non-comment lines in the file clear the R workspace (memory) and set the working directory(folder) to the folder where you have the exercises. ***You will need to modify the working directory***. Execute these lines in R.

```
rm(list=ls())                          #   clear workspace
setwd('h:/x/workshops/uf2016/exercises')    #   set working directory(folder)
library(RMark)
```

2) Assuming that you have already created a MARK input file, the first step in using RMark is to convert the input file into an RMark input "data-frame". (You can think of a data-frame as a spreadsheet in R, where you can access the spreadsheet by rows and columns, or by variable(column) name. This conversion is done with the "convert.inp" function. The only required argument is the input filename (mp2age.inp), but since we have 4 groups in our input file, we also need to specify the group names to the convert.inp function. This is done with the group.df argument. The converted data-frame is saved with the name, "mpinp". Execute that line, then type the new variable name into the R window.

```
mpinp=convert.inp('mp2age.inp',
  group.df=data.frame(agegrp=c('a','a','y','y'),sex=c('F','M','F','M'))
)
Mpinp
```

Notice that I didn't name the age-group variable, "age". This is intentional and due to the fact that RMark has a pre-defined variable named "age". The RMark variable, "age" can be used for certain things, but not in this case where we want to classify animals into two groups: young at first capture and adult at first capture. We'll use the "age" variable to make models where animals can be "young" in the 1st occasion, then adult after.

3) The 2nd step is to create a processed-data variable which contains other variables needed to setup and run the MARK models. The variable, "mppr" is a "list" variable, or a variable which contains other variables. The RMark function to do this is "process.data" and requires the converted input from the previous step, the type of MARK models which will be run, and the group variable name (if applicable). Execute this line, then type 'mppr' to see the contents of this variable.

```
mppr = process.data(mpinp, model='CJS', groups=c("agegrp","sex"))
```

4) The 3rd step is to create design-matrix data variables, needed by MARK to build models. The RMARK function is “make.design.data” and the processed-data variable created in the previous step is needed as an argument to the function. Execute this line and type ‘mpdd’ to view the contents of this variable. This is a list-type variable and it contains a data-frame variables, Phi (for apparent survival) and p (capture probability) for the estimated parameters for the model-type (CJS) we specified. The columns of Phi and p are the variables we can use in building MARK models.

```
mpdd = make.design.data(mppr)
mpdd$Phi
```

5) In order to build models where animals can be “young” for 1 capture occasion and “adult” for other occasions, we need to add a variable to our design-data variable. We’d like to create a new variable for Phi which takes on two values:

   0 if animal is in group, “young” AND months-since-orig-capture <= 0

   1 otherwise

```
mpdd$Phi$agecl=1  #   create new variable, agecl
#       next, get row numbers where group=”y” and age=0
i=((substr(mpdd$Phi$group,1,1)=="y") & (mpdd$Phi$Age==0))
mpdd$Phi$agecl[i]=0  #  set new variable to zero for those rows
print(mpdd$Phi)  #  look at new Phi data-frame
```

Modeling strategy is to develop models corresponding to our hypotheses of how survival and/or capture probabilities are affected by age, sex and time.

6) Now we’re ready to build our first MARK model. We’ll start with the most simple model (survival constant over time and equal among age/sex groups). To create a model, we call the ‘make.mark.model’ function with the processed-data variable created in step 3 (mppr), the design-data variable created in step 4 (mpdd), title, and list of parameters as arguments. Here, formula=~1 means the parameter is a constant value. Execute these 3 lines and type ‘phi_1_p_1’ to examine what we’ve created. This variable is another list-variable which contains other variables to be used when we run the model.

```
phi_1_p_1 = make.mark.model(mppr,mpdd,title='Patuxent Mp data',parameters=list(
  Phi=list(formula=~1),
  p=list(formula=~1)
))
```

7) We can run the model by calling the 'run.mark.model' function with the model variable created in step 6 as the argument. Execute this line then type 'phi_1_p_1_out' to examine the contents. The output from MARK is stored in a text file and by typing 'phi_1_p_1_out', the text file is opened in notepad. This is the usual output you would get if you ran MARK interactively using its GUI. The output is also stored as an R list-variable. Type 'str(phi_1_p_1_out)' to display the structure of this list-variable. What is the survival rate estimate from this model? You can get it from the text output displayed in notepad, or by looking at the phi_1_p_1_out$results$real variable.

```
phi_1_p1_out=run.mark.model(phi_1_p_1)
Phi_1_p_1_out$results$real
```

8) To run other models, we only need to repeat the last 2 steps: create.mark.model, and run.mark.model. With these data, the interest was in whether survival and/or capture probabilities were different among age and sex classes, as well as time. So, we make new mark models, with different assumptions about survival and capture probabilities. Execute these lines, as well as the line to run the models.

```
#           make model with age-specific survival, phi(a)p(.)
phi_a_p_1 = make.mark.model(mppr,mpdd,parameters=list(
 Phi=list(formula=~agecl),
 p=list(formula=~1)
))
phi_a_p_1_out = run.mark.model(phi_a_p_1)

#           make model with age and sex-specific survival, phi(a*s)p(.)
phi_axs_p_1 = make.mark.model(mppr,mpdd,parameters=list(
 Phi=list(formula=~agecl*sex),
 p=list(formula=~1)
))
phi_axs_p_1_out = run.mark.model(phi_axs_p_1)

#                 make model with age and sex-specific survival, sex-specific capt. probs,
phi(a*s)p(s)
phi_axs_p_s = make.mark.model(mppr,mpdd,parameters=list(
 Phi=list(formula=~agecl*sex),
 p=list(formula=~sex)
))
```

```
phi_axs_p_s_out = run.mark.model(phi_axs_p_s)
#                    make model with age,sex,time-specific survival, sex-specific capt. probs,
phi(a*s*t)p(s)
phi_axsXt_p_s = make.mark.model(mppr,mpdd,parameters=list(
 Phi=list(formula=~agecl*sex*time),
 p=list(formula=~sex)
))
phi_axsXt_p_s_out = run.mark.model(phi_axsXt_p_s)

#                      make model with age,time-specific survival, sex-specific capt. probs,
phi(a*t)p(s)
phi_aXt_p_s = make.mark.model(mppr,mpdd,parameters=list(
 Phi=list(formula=~agecl*time),
 p=list(formula=~sex)
))
phi_aXt_p_s_out = run.mark.model(phi_aXt_p_s)

#            make model with age,time-specific survival, sex-specific capt. probs
#            with additive effect of age on survival
phi_aPt_p_s = make.mark.model(mppr,mpdd,parameters=list(
 Phi=list(formula=~agecl+time),
 p=list(formula=~sex)
))
phi_aPt_p_s_out = run.mark.model(phi_aPt_p_s)
```

9) We can create a table of model results by calling the 'model.table' function, with the names of the variables which contain the output of each model as an argument. Execute this line, and the next line to print the table.

```
tbl=model.table(model.list=ls(pattern="phi.+out"),adjust=T,use.lnl=T)
print(tbl)
```

Shortcut: Instead of listing all of the models by name in the model list, I used the R function, "ls", which lists all variables in the workspace which start with "phi" and end with "out".

Questions:

1. The models with interaction between age and time and additive time + age effects were both competitive. In the interactive model, adult survival is higher in

some months and young survival in others. Can this happen as well in the additive model? Why or why not?

2. In many species of vertebrates, young are predicted to have lower apparent survival than adults. Was this true in the example? What biological stories might explain the direction of the estimated average difference between young and adult survival?
3. The a priori hypothesis was that males would have higher capture probabilities than females. Was this true? What biology might underlie this prediction and difference?
4. Some population models (such as stochastic projection matrices) require estimates of true temporal variance of vital rates such as survival. But what other source of variation is present in the monthly variation among survival estimates? How can the true temporal variance be separately estimated?
5. The CJS model permits inference about capture and apparent survival probabilities, as shown. But under the JS model, we can also estimate abundance of adults. How do we do this?
6. Why can't we estimate the number of young in the same manner as for adults? What piece of information are we missing?

Exercise 3: The Multistrata Model

This exercise closely follows the content of the 6th lecture and is mostly intended to show how to run program MARK to compute survival, transition and capture probabilities from 'capture-recapture' data.. We have prepared exercises for use in either MARK or RMark.

Data input

Background
Data for this example came from the trapping of meadow voles, *Microtus pennsylvanicus*, at Patuxent Wildlife Research Center, Laurel, MD (Nichols *et al.*, 1984). Data were collected on a 10 x 10 grid of trapping stations spaced at 7.6m intervals in old field habitat. A single modified Fitch live trap (Rose, 1973) was placed at each station. Hay and dried grass were placed in the traps and whole corn was used as bait. Sampling occurred for five consecutive days each month, from June 1981 through December 1981. During each 5-day trapping session, traps were opened in the evening of the first day, checked the following morning, locked open during the day, and reset in the evening, with the sequence repeated each day until 5 days had elapsed. A racoon, *Procyon lotor* (later captured), visited the traps on the final two nights of the second trapping session, essentially leaving only 3 days of trapping for this session. At each capture, animals were examined for a tag, sexed, weighed, and examined for external reproductive characteristics. Tagged animals were eartagged with numbered fingerling tags, and tag numbers of marked animals were recorded at each capture.

We used 'adult' (>22g) animals and collapsed the 5 days of sampling each month into a single assessment of presence or absence, leaving 11 monthly sampling occasions. For each capture, the location of capture was recorded. Individuals captured in locations with X-coordinate in the range 1-5, were assigned capture-code ‘1’. Those captured in locations with X-coordinate in the range 6-10 were assigned capture-code ‘2’.

For R users

1) Open the file “multistrata.r” in your preferred text editor for R. **You will need to modify the working directory**. Execute the lines to clear the workspace, set the working directory and load the RMark library.

```
rm(list=ls()); setwd('h:/x/workshops/uf2016/exercises/ex3_multistrata_mp')
library(RMark)
```

2) The first step is to convert the input file into an RMark input “data-frame. This conversion is done with the “convert.inp” function. The only required argument is the input filename (mp2age.inp), but since we have 2 groups in our input file, we also need to specify the group names to the convert.inp function. This is done with the group.df argument. The converted data-frame is saved with the name, “mpinp”. Execute that line, then type the new variable name into the R window.

```
mpinp = convert.inp('multistrata.inp',group.df=data.frame(sex=c('F','M')))
```

3) The 2nd step is to create a processed-data variable which contains other variables needed to setup and run the MARK models. The RMark function to do this is “process.data” and requires the converted input from the previous step, the type of MARK models which will be run, and the group variable name (if applicable). Execute this line, then type ‘mppr’ to see the contents of this variable.

```
mppr = process.data(mpinp, model='Multistrata', groups="sex")
```

4) The 3rd step is to create design-matrix data variables, needed by MARK to build models. The RMARK function is “make.design.data” and the processed-data variable created in the previous step is needed as an argument to the function. Execute this line.

```
mpdd = make.design.data(mppr)
```

Modeling strategy is to develop models corresponding to our hypotheses of how survival, transition and/or capture probabilities are affected by sex and time. One of the primary objectives of the study was to determine the effects of habitat fragmentation. Just after the 4th month, a strip was mowed down the middle of the area. The hypothesis was that the mowing of the strip would cause movement of animals between the two grid halves to decrease after the 4th month.

5) Now we're ready to build our first MARK model. We'll start with the most simple model (survival transition and capture probabilities constant over time). To create a model, we call the 'make.mark.model' function with the processed-data variable created in step 3 (mppr), the design-data variable created in step 4 (mpdd), title, and list of parameters as arguments. As a reminder, the formula, "~1" means that the parameter is constant over all values of time and group (sex).

```
s_1_psi_1_p_1 = make.mark.model(mppr,mpdd,title='Patuxent MP data',parameters=list(
 S=list(formula=~1),
 Psi=list(formula=~1),
 p=list(formula=~1)
))
```

6) We can run the model by calling the 'run.mark.model' function with the model variable created in step 5 as the argument. Execute this line then type 's_1_psi_1_p_1_out' to examine the contents. The output from MARK is stored in a text file and by typing 's_1_psi_1_p_1_out', the text file is opened in notepad. This is the usual output you would get if you ran MARK interactively using its GUI. The output is also stored as an R list-variable. Type 'str(s_1_psi_1_p_1_out)' to display the structure of this list-variable.

```
s_1_psi_1_p_1_out=run.mark.model(s_1_psi_1_p_1)
```

7) To run other models, we only need to repeat the last 2 steps for each model. Run the next model by executing the appropriate lines in the file.

```
s_t_psi_t_p_t = make.mark.model(mppr,mpdd,parameters=list(
 S=list(formula=~time),
 Psi=list(formula=~time),
 p=list(formula=~time)
))
s_t_psi_t_p_t_out = run.mark.model(s_t_psi_t_p_t)
```

8) To make a model where transition rates were one value before the mowing of the strip, and another value after the mowing, we'll need to create a new design-data variable. We'll call it "twoper" and set it to zero for the 1st 4 months and one for the last 6 months.

```
mpdd$Psi$twoper=0
mpdd$Psi$twoper[mpdd$Psi$Time>3]=1
```

9) We can now build and run our model where transition rates are one value for the 1st 4 months and another value after.

```
s_t_psi_2per_p_t = make.mark.model(mppr,mpdd,parameters=list(
 S=list(formula=~time),
 Psi=list(formula=~twoper),
 p=list(formula=~time)
))
s_t_psi_2per_p_t_out = run.mark.model(s_t_psi_2per_p_t)  #        run model
```

10) We can create a table of model results by calling the 'model.table' function, with the names of the variables which contain the output of each model as an argument. Execute this line, and the next line to print the table.

```
tbl=model.table(model.list=ls(pattern="s_.+out"),type="Known")
print(tbl)
```

Questions:

(1) This grid was part of an experiment designed to test hypotheses about effects of fragmentation on meadow vole population dynamics. If the fragmentation created by the strips of bare ground really affected movement, what predictions would we make about effects of fragmentation on the 3 sets of model parameters (survival, capture and movement probabilities)?

(2) Based on AIC, which model appears to be best supported by the data? What conclusions can you draw from this experiment based on the AIC table? Are the parameter estimates themselves relevant to conclusions or does AIC provide all of the information that you need?

(3) Have a look at the estimates of movement probabilities from the 2 time periods, before and after fragmentation. Are they consistent with your predictions?

(4) Do the results from this grid provide strong inferences about effects of fragmentation? If not, what other information would be useful in strengthening the inferences?

(5) In (1) did you make any predictions about changes in survival probability associated with fragmentation? If so, what was your rationale? The top model provided time-specific estimates of survival, so we could compute and compare means for the periods before and after fragmentation. Another way to obtain inference about this contrast is to look at estimates arising from the 3rd model, in which survival is computed for 2 time periods (before and after fragmentation). What do these look like? Are they consistent with predictions?

Heterogeneity and Exploitation: Exercise 11

Uses spreadsheet file, “Ex11_het_surv_harv.xlsx”

For the purpose of this exercise, we will define “demographic heterogeneity” as variation among individuals of a population in vital rates (survival or reproduction) that is not associated with any visible trait of the organism. For example, individuals within our population may vary in underlying survival probabilities (perhaps we have a high-survival group and a low-survival group), but there is no morphological or behavioral characteristic that we can observe that will allow us to categorize individuals. So the variation is invisible to us. This exercise deals with heterogeneity in underlying survival probabilities, although heterogeneity in reproductive rates is possible (and likely) as well.

We will begin with a harvested population that includes 2 groups of individuals, differing in survival only. We will consider an annual anniversary date at the beginning of the hunting season and apply a harvest rate (*h*) during the hunting season (keep things simple and assume no non-harvest mortality during the hunting season). The survivors of the hunting season then experience a probability of surviving nonhunting mortality sources (*S*). Because only hunting occurs during the hunting season portion of the year and only nonhunting mortality occurs during the remainder of the year we can obtain total annual survival (S_{Tot}) as the product of these *net* survival rates [$S_{Tot} = S(1\text{-}h)$]. Recall that *net* rates associated with a specific mortality source are finite rates that would occur in the absence of any other mortality source,

Take the following numerical example, beginning with 1000 individuals in each group:

Group 1 (low survival): $S = 0.4$, $h = 0.2$

Group 2 (high survival): $S = 0.6$, $h = 0.1$

Before using the spreadsheet, answer the following questions:

1. What is the annual survival rate for each individual within each group? Does the example make sense? In other words, if such heterogeneity exists, is it sensible that the individuals with the lower probabilities of surviving nonhunting mortality also have lower probabilities of surviving hunting mortality. What kinds of biological stories might underlie such variation?
2. We specified no time variation, so annual survival rates should be constant within each group. But what about overall survival of this 2-group mixture (i.e., what about annual survival rate of the combined groups). Should this also be constant over time? Why or why not? Assume that this example represented a cohort of young animals in year 1, aging and eventually reaching age 10. If we had no knowledge of the 2-group mixture, what might we conclude about age-specificity of survival?
3. Similar question to above. Harvest rates within each group are specified as constant over time (e.g., we apply the same set of harvest regulations each year). Should the overall harvest rate of the 2-group mixture be constant as well? Why or why not?
4. Use the spreadsheet, plug in the above survival and harvest rates and observe the trajectories of 2-group survival and harvest over time. If the trajectories did not correspond to your answers above, can you now explain exactly what happened?
5. So how might such heterogeneity be relevant to harvest theory? How might it be relevant to PVAs?
6. What would you expect to happen if you increase (make the rate differences larger) or decrease the variation between the 2 groups? Test your expectation with a couple of examples using the spreadsheet.
7. What happens when you reverse the relationship between nonhunting survival and harvest rates (i.e., low nonhunting survival corresponds to low harvest rate and high nonhunting survival corresponds to high harvest rate)?

Exercise 13: Direct Estimation of Lambda

This exercise closely follows the content of the 6th lecture and is mostly intended to show how to run program MARK to compute estimates of lambda 'known-fate' data.. We have prepared exercises for use in either MARK or RMark.

Data input

Input data consists of individual capture-histories.

The example we'll use for this model contains 2 groups of animals: Males and Females. So, each capture-history will have 2 frequencies following the encounter-history. Since the data are not summarized, one of the frequencies will be zero and one will be one. The following is a sample MARK input file (nwc_lamb.inp):

```
/*1177-06702 M A*/00000000000000111111110111 1 0;
/*1177-06708 F A*/00000000000011111111111100 0 1;
/*1177-06724 M A*/00000000000011111111111100 1 0;
/*1177-06728 M A*/00000000000001111111110000 1 0;
/*1177-06730 F S1*/00000000000001000000000000 0 1;
/*1177-06731 F S1*/00000000000001110000000000 0 1;
/*1177-06732 M S2*/00000000000000011111111111 1 0;
/*1177-06737 M A*/00000000000001110000000000 1 0;
/*1177-06740 F S1*/00000000000001000000000000 0 1;
```

Note: Comments can appear in the input file, surrounded by “/*” and “*/”.

For R users

1) Open the file “ex13_direct_est_lambda.r” in your preferred text editor for R. **You will need to modify the working directory**. Execute the lines which clear the workspace, set the working directory and load the RMark library.

   ```
   rm(list=ls()); setwd('h:/x/workshops/uf2016/exercises/ex13_direct_est_lambda')
   library(RMark)
   ```

2) The first step in using RMark is to convert the input file into an RMark input “data-frame”. This conversion is done with the “convert.inp” function. The only required argument is the input filename (nwc_lamb.inp), but since we have 2 groups in our input file, we also need to specify the group names to the convert.inp function. This is done with the group.df argument. The converted data-frame is saved with the name, “owlinp”. Execute that line, then type the new variable name into the R window.

   ```
   owlinp = convert.inp('nwc_lamb.inp',group.df=data.frame(sex=c('M','F')))
   ```

3) The 2nd step is to create a processed-data variable which contains other variables needed to setup and run the MARK models. The RMark function to do this is “process.data” and requires the converted input from the previous step, the type of MARK models which will be run, and the group variable name (if applicable). Execute this line, then type ‘owlpr’ to see the contents of this variable.

   ```
   owlpr = process.data(owlinp,model='Pradrec',groups="sex")
   ```

4) The 3rd step is to create design-matrix data variables, needed by MARK to build models. The RMARK function is “make.design.data” and the processed-data variable created in the previous step is needed as an argument to the function. Execute this line and type ‘owldd’ to view the contents of this variable. This is a list-type variable and it contains a data-frame variable (S) for the estimated parameter for the model-type (Known) we specified. The columns of S are the variables we can use in building MARK models.

   ```
   owldd = make.design.data(owlpr)
   ```

   Modeling strategy is to develop models corresponding to our hypotheses of how survival, fecundity and/or capture probabilities are affected by sex and time.

5) Now we’re ready to build and run our MARK models. Instead of specifying each model and running it, we will create a model structure for each parameter which corresponds to our

belief of what makes the parameter vary. For example, we believe that survival (Phi) might be constant or might be time-specific. So, we create two variables which contain a formula for a possible model:

```
Phi.dot=list(formula=~1)
Phi.t=list(formula=~time)
```

Next, we create two more new variables containing formulae for our belief about capture probabilities:

```
p.dot=list(formula=~1)
p.t=list(formula=~time)
p.s=list(formula=~sex)
```

And two variables containing formulae for our belief about how fecundity might vary:

```
f.dot=list(formula=~1)
f.t=list(formula=~time)
```

6) With these formula variables, we can create a list of all possible combinations of those parameters and formula using the “create.model.list” function.

   ```
   mod.list=create.model.list("Pradrec")
   ```

7) Now, we can run all models in this list using the “mark.wrapper” function. This function runs each model in the model list and saves the output of each model as a list variable.

   ```
   mod.out=mark.wrapper(mod.list,data=owlpr,ddl=owldd)
   ```

8) The “mark.wrapper” function automatically creates a results AIC table, as well as storing the results of each model. We can obtain the model table from the output variable, “mod.out” by selecting the sub-variable, “model.table” from the list variable, “mod.out”:

   ```
   tbl=mod.out$model.table
   ```

   and print the table (without 1st 3 columns) with:

   ```
   cat('\nTable of model results:\n')
   ```

```
print(tbl[,-1:-3,])   #  print table (without redundant columns 1 to 3)
```

Questions:

1. **Why do you suppose that scientists shifted from the use of matrix models to direct estimation of λ_t?**
2. **Which model would you choose to best describe these data? Why?**
3. **Models included time-specific variation for survival and recruitment. Based on general principles of evolutionary ecology, which of these parameters would you think most likely to exhibit year-to-year variation? Does the evidence provided by model selection agree with this prediction?**
4. **In the model with all parameters time-specific, are there any parameter estimates that you would view as unusable? If so, which estimates would you not use? Why?**
5. **So how are owls on this study area doing? Compute the proportional change in population size between years 4 and 23. By what fraction has the population grown or declined?**

Exercise 13: Contributions to Lambda

This exercise closely follows the content of the 6th lecture and is mostly intended to show how to run program MARK to compute seniority and capture probabilities from 'capture-recapture' data. We have prepared exercises for use in RMark.

Data input

Background
Data for this example came from the trapping of meadow voles, *Microtus pennsylvanicus*, at Patuxent Wildlife Research Center, Laurel, MD (Nichols *et al.*, 1984). Data were collected on a 10 x 10 grid of trapping stations spaced at 7.6m intervals in old field habitat. A single modified Fitch live trap (Rose, 1973) was placed at each station. Hay and dried grass were placed in the traps and whole corn was used as bait. Sampling occurred for five consecutive days each month, from June 1981 through December 1981. During each 5-day trapping session, traps were opened in the evening of the first day, checked the following morning, locked open during the day, and reset in the evening, with the sequence repeated each day until 5 days had elapsed. A racoon, *Procyon lotor* (later captured), visited the traps on the final two nights of the second trapping session, essentially leaving only 3 days of trapping for this session. At each capture, animals were examined for a tag, sexed, weighed, and examined for external reproductive characteristics. Tagged animals were ear tagged with numbered fingerling tags, and tag numbers of marked animals were recorded at each capture.

We used 'adult' (>22g) animals and collapsed the 5 days of sampling each month into a single assessment of presence or absence, leaving 6 monthly sampling occasions.

```
# Data input
#   Input data consists of summarized frequencies of capture-histories.
#
#  The example we'll use for this model  contains 2 groups of animals:  Males and
```

# females, initially captured as adults. So, each capture-history will
# have 2 frequencies following the capture-history. The input file has already
# been created and is named: 'mp1age.inp'.
#
100000   7   8 /* F   M */;
100000 -1 -3;
110000 10 21;
110000 -4 -2;
111000   7   5;
111100   2   3;
111100   0 -1;
:   :   :

# Explanation: The 1st line indicates that 7 female and 8 male individuals were captured only in
#   time period 1 and were released. The 2nd line indicates that 1 female and 3 male
#   individuals were captured onl in time period 1 and not released ("-" indicates
#   not released).

# Here are the steps to run MARK on this input file:

```
rm(list=ls())                #   clear workspace
library(RMark)              #   add RMark functions to our R workspace
```

#   convert MARK input file to RMARK data frame, while defining the 2 groups

```
mpinp =
convert.inp('mp1age.inp',group.df=data.frame(sex=c('F','M')),use.comments=F)
```

#   process data frame, specifying model type
#   (Pradrec=Pradel model, recruitment parameterization)

```
mppr = process.data(mpinp, model='Pradel', groups=c("sex"))
```

#   make design matrix data variables from processed data frame

```
mpdd = make.design.data(mppr)
```

# Modeling strategy is to develop a pre-defined set of models. Each model represents a
# plausible hypothesis about survival or capture probabilities of the animals.

```
Gamma.dot=list(formula=~1)
Gamma.s=list(formula=~sex)
Gamma.t=list(formula=~time)
Gamma.sXt=list(formula=~sex*time)
Gamma.sPt=list(formula=~sex+time)
p.dot=list(formula=~1)
p.s=list(formula=~sex)
p.t=list(formula=~time)
p.sXt=list(formula=~sex*time)
p.sPt=list(formula=~sex+time)

mod.list=create.model.list("Pradel")
```

#   create AIC table of model results for model comparison

```
mod.out=mark.wrapper(mod.list,data=mppr,ddl=mpdd)
tbl=mod.out$model.table
```

# print table (without redundant columns 1 & 2)

```
print(tbl[,-1:-2,])
```

# print/plot seniority estimates from top model...

```
i=as.numeric(rownames(tbl))[1]  #  get top model number from table
cat('\nEstimates from top model:',as.character(tbl$model[1]),'\n')
print(mod.out[[i]]$results$real)  #  print estimates from top model

plot(1:5,mod.out[[i]]$results$real$estimate[1:5],type='b',ylim=c(0,1),
     main=as.character(tbl$model[1]),xlab='time',ylab='seniority')
```

Questions:

(1) Define in words the probability estimated by the seniority parameter, gamma.

(2) In this example, do the gammas suggest that survival (of previous members of the population) or recruitment (of new members) make the larger contributionto population growth?

(3) This vole example is from Maryland and uses monthly intervals between trapping sessions. If you had to guess, during what season of the year do you think the last 2 months occurred?

(4) The various methods that we have discussed for making inferences about lambda and contributions to lambda are all interrelated. Say we used Pradel's full temporal symmetry model for inference about lambda, but used the phi(i), f(i) parameterization (survival and per capita recruitment). How could you use these parameters to estimate the same relative contributions provided by the gammas?